UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/051,274	01/22/2002	Sunil Kunisetty	50277-1020	7586

7590 06/17/2005	EXAMINER
DITTHAVONG & CARLSON, P.C.	PHAM, CHRYSTINE

ART UNIT	PAPER NUMBER
2192	

DITTHAVONG & CARLSON, P.C.
Suite A
10507 Braddock Rd
Fairfax, VA 22032

DATE MAILED: 06/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/051,274	KUNISETTY ET AL.
	Examiner	Art Unit	
	Chrystine Pham	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on _07 April 2005_.

2a) ☒ This action is **FINAL.**   2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) _1-14_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) _1-14_ is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage

        application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1.   This action is responsive to Amendment filed on April 7th 2005. Claim 8 has been amended.

Claims 9-14 are new claims. Claims 1-14 are presented for examination.

Response to Arguments

2.   Applicant's arguments filed April 7th 2005 have been fully considered but they are not persuasive.

        The Applicants essentially contend that Friesen does not teach the "shared, read-only

memory" recited in independent claims 1, 5, and 8. More specifically, the Applicants rely on

col.3:28-30 to assert that "Friesen's self-identified key technique does not work with shared, read-

only memory" (pages 8-9). In other words, the Applicants attempt to equate Friesen's "modifying

static variables during execution of the program" with not teaching "a shared, read-only memory".

The Examiner firmly disagrees and submits that the static variables in col.3:28-30 refer to static

variables for a shopping cart discussed in the same paragraph. It is further submitted that the

"shopping cart" frame (i.e., ShopFrame) is only an <u>object</u> or an <u>instance</u> of a **class** implementing

the "shopping cart" object (see at least *class app, class ShopFrame, class ShopPanel* col.15:15-

col.16:61). Thus, modifying values for static variables in a shopping cart frame or instance (e.g.,

user deselects an item, adds a new item, or changes quantity of a selected item) has no effect on

the **class** containing the defined static variables and does not equate to modifying the **class**

containing the defined static variables. In col.9:48-col.10:46, Friesen discloses a database or

cache "update.txt" containing an itemPrice for its corresponding itemID. Friesen explicitly states

that this cache is defined as a static String variable in a class implementing the shopping cart

object. In col.19:35-50, Friesen explicitly discloses Constants.class (which implements the

shopping cart object) containing said static String variable. It is inherent that the static String is

pre-initialized to "update.txt" in said class. It is further inherent and essential to Friesen's

invention that said class is loaded into a shared, read-only memory. Since if the class is not in

the shared, read-only memory, itemPrice can be freely modified for any corresponding itemID.

Furthermore, Friesen's invention is directed persisting a shopping cart object from one web page

to another. It is inherent that classes (containing static variables) implementing the shopping cart

object are persisted in memory (i.e., **read-only**) from one web page to another (i.e., **shared** by

web pages), thus enables Friesen to eliminate the need to download the same classes (in order

to maintain the shopping cart object) every time the user accesses a different web page.

3.   In view of the foregoing discussion, rejection of claims under 35 USC 102(e) is considered proper

and maintained.

Claim Rejections - 35 USC § 102

4.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for

the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5.   Claims 1-14 are rejected under 35 U.S.C. 102(e) as being anticipated by Friesen (US 6636863),

hereinafter, *Friesen.*

Claim 1

Friesen teaches a method of compiling a page containing markup text (e.g., see *web page 10*

FIG.1 & associated text) into an application (e.g., see *shopping cart frame 12* FIG.1 & associated text;

see *JAVA applet, Active X object* col.5:35-67; see *code, frame, web page* col.11:30-col.12:40) that

outputs markup (e.g., see *shopping cart frame 12* FIG.1 & associated text; col.5:35-47; see col.6:60-

col.7:10; see *applet* col.15:10-45; see *frame* col.16:19-20) in response to a request from a user (e.g.,

see *items for sale, selected, added, purchaser, decreasing the quantity of an item, submitting a final order*

col.5:23-47; see *operator, finalize a purchase* col.6:50-60; see *workstation 36* FIG.2 & associated text;

see *request* col.17:37-42), said method comprising:

    o   pre-initializing a static variable of a class to contain the markup text (e.g., see *shopping cart,*

        identities, prices, static variables col.3:30-40; see *class ShopPanel, variables* col.15:10-col.16:6;

        see *app.class, Constants.class* col.17:37-52; see *Constants.class, Vector class, item ID, item*

        price col.18:40-50; see *static String variable* col.19:48-50); and

    o   loading the class containing the pre-initialized static variable into a shared, read-only

        memory (e.g., see *programs, web pages, memory, subsequent use, shopping cart, first web*

        page col.2:67-col.3:15; see *update.txt, 'itemPrice', cache* col.10:11-37; see *Constants.class,*

        Vector class, item ID, item price col.18:40-50; see *static String variable* col.19:48-50; *itemID,*

        itemPrice, database, cache, "update.txt, static String, JAVA class col.9:48-col.10:45).

Claim 2

The rejection of base claim 1 is incorporated. *Friesen* further teaches storing the markup text in a

resource file associated with the application (e.g., see *MICROSOFT ACCESS, update.txt* col.10:10-29;

see *update.txt, Constants.class* col.19:48-50).

Claim 3

The rejection of base claim 1 is incorporated. *Friesen* further teaches wherein the step of pre-

initializing the static variable includes the step of:

    o   reading the markup text from the resource file (e.g., see *programs, web pages, memory,*

        subsequent use, shopping cart, first web page col.2:67-col.3:15; see *shopping cart, static*

        variables col.3:30-40; see *frame, static variables, ShopPanel, shopping cart* col.16:19-61; see

        size, sales Vector, Constants.class col.18:60-65); and

   o   initializing the static variable of the class based on the read markup text (e.g., see *shopping*

cart, static variables col.3:30-40; see *frame, static variables, ShopPanel, shopping cart* col.16:19-

61; see *instantiate the shopping cart frame* col.18:60-65).

Claim 4

The rejection of base claim 1 is incorporated. *Friesen* teaches a computer-readable medium (e.g.,

see *memory 44* FIG.2 & associated text) bearing instructions (e.g., see *browser application 50* FIG.2 &

associated text) that, when executed, cause one or more processors (e.g., see *software engine 52,*

processor 40 FIG.2 & associated text; see *JVM* col.6:25-40) to perform the method according to claim 1

(e.g., see *software engine 52, processor 40* FIG.2 & associated text; see *JVM, programs, applets, web*

pages col.6:25-46).

Claim 5

Friesen teaches a method of compiling a page containing markup text (e.g., see *web page 10*

FIG.1 & associated text) into an application (e.g., see *shopping cart frame 12* FIG.1 & associated text;

see *JAVA applet, Active X object* col.5:35-67; see *code, frame, web page* col.11:30-col.12:40) that

outputs markup (e.g., *shopping cart frame 12* FIG.1 & associated text; col.5:35-47; see col.6:60-col.7:10;

see *applet* col.15:10-45; see *frame* col.16:19-20) in response to a request from a user (e.g., see *items*

for sale, selected, added, purchaser, decreasing the quantity of an item, submitting a final order col.5:23-

47; see *operator, finalize a purchase* col.6:50-60; see *workstation 36* FIG.2 & associated text; see *request*

col.17:37-42), said method comprising:

   o   generating instructions for the application (e.g., see _browser application 50 FIG.2 &

associated text), borne on a computer-readable medium (e.g., see *memory 44* FIG.2 &

associated text), said instructions that, when executed, cause one or more processors to

perform (e.g., see *software engine 52, processor 40* FIG.2 & associated text; see *JVM* col.6:25-

40; see *software engine 52, processor 40* FIG.2 & associated text; see *JVM, programs, applets, web pages* col.6:25-46) the steps of:

o   in one time the application is executed, loading the class containing a static variable into a shared, read-only memory, said static variable being pre-initialized to contain the markup text (e.g., see *programs, web pages, memory, subsequent use, shopping cart, first web page* col.2:67-col.3:15; see *update.txt, 'itemPrice', cache* col.10:11-37; see *Constants.class, Vector class, item ID, item price* col.18:40-50; see *static String variable* col.19:48-50; see *shopping cart, identities, prices, static variables* col.3:30-40; see *class ShopPanel, variables* col.15:10-col.16:6; see *app.class, Constants.class* col.17:37-52; see *Constants.class, Vector class, item ID, item price* col.18:40-50; see *static String variable* col.19:48-50; *itemID, itemPrice, database, cache, "update.txt, static String, JAVA class* col.9:48-col.10:45); and

o   in subsequent time the application is executed, accessing the markup text in the shared, read-only memory (i.e., <u>class</u> is <u>not loaded</u> into the <u>shared, read-only memory</u> in the <u>subsequent time</u> the application is executed) (e.g., see *programs, web pages, memory, subsequent use, shopping cart, first web page* col.2:67-col.3:15; see *shopping cart, static variables* col.3:30-40; see *frame, static variables, ShopPanel, shopping cart* col.16:19-61; see *size, sales Vector, Constants.class* col.18:60-65);

Claim 6

The rejection of base claim 5 is incorporated. Claim recites limitations, which have been addressed in claim 5, therefore, is rejected for the same reasons as cited in claim 5.

Claim 7

Friesen teaches a computer-readable medium (e.g., see *memory 44* FIG.2 & associated text) bearing instructions (e.g., see *browser application 50* FIG.2 & associated text) that, when executed,

cause one or more processors (e.g., see *software engine 52, processor 40* FIG.2 & associated text; see

JVM col.6:25-40) to perform the method according to claim 5 (see claim 5).

Claim 8

Friesen teaches a computer-readable medium (e.g., see *memory 44* FIG.2 & associated text)

bearing instructions for producing markup based on static markup text (e.g., see *browser application*

50 FIG.2 & associated text; see *web page 10* FIG.1 & associated text; see *shopping cart frame 12* FIG.1

& associated text; col.5:35-47; see col.6:60-col.7:10; see *applet* col.15:10-45; see *frame* col.16:19-20)

said instruction arranged, when executed, to cause one or more processors (e.g., see *software engine*

52, processor 40 FIG.2 & associated text; see *JVM* col.6:25-40) to perform the steps of:

- o in one time the application is executed, loading a class containing a pre-initialized static

   variable into a shared, read-only memory, said static variable being pre-initialized to contain

   the markup text (e.g., see *programs, web pages, memory, subsequent use, shopping cart, first*

   web page col.2:67-col.3:15; see *update.txt, 'itemPrice', cache* col.10:11-37; see *Constants.class,*

   Vector class, item ID, item price col.18:40-50; see *static String variable* col.19:48-50; see

   shopping cart, identities, prices, static variables col.3:30-40; see *class ShopPanel, variables*

   col.15:10-col.16:6; see *app.class, Constants.class* col.17:37-52; see *Constants.class, Vector*

   class, item ID, item price col.18:40-50; see *static String variable* col.19:48-50; *itemID, itemPrice,*

   database, cache, "update.txt, static String, JAVA class col.9:48-col.10:45); and

- o in subsequent time the application is executed, accessing the markup text in the shared,

   read-only memory (e.g., see *programs, web pages, memory, subsequent use, shopping cart,*

   first web page col.2:67-col.3:15; see *shopping cart, static variables* col.3:30-40; see *frame, static*

   variables, ShopPanel, shopping cart col.16:19-61; see *size, sales Vector, Constants.class*

   col.18:60-65);

Claim 9

The rejection of base claim 1 is incorporated. *Friesen* further teaches wherein the markup text includes information to be displayed to a user an annotation directing a user agent how to render the information to be displayed to the user; and the markup output by the application includes the annotation (e.g., see *shopping cart frame 12* FIG.1 & associated text; col.5:35-47; see col.6:60-col.7:10; see *applet* col.15:10-45; see *frame* col.16:19-20).

Claim 10

The rejection of base claim 1 is incorporated. *Friesen* further teaches wherein the static variable of a class is an array of characters (see at least *static String variable* col.19:48-50; *itemID, itemPrice, database, cache, "update.txt, static String, JAVA class* col.9:48-col.10:45).

Claims 11, 13

Claims recite limitations, which have been addressed in claim 9, therefore, therefore, are rejected for the same reasons as cited in claim 9.

Claims 12, 14

Claims recite limitations, which have been addressed in claim 10, therefore, are rejected for the same reasons as cited in claim 10.

Conclusion

6.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be
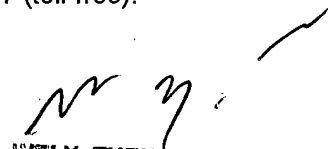
calculated from the mailing date of the advisory action. In no event, however, will the statutory

period for reply expire later than SIX MONTHS from the mailing date of this final action.

7.   Any inquiry concerning this communication or earlier communications from the examiner should

be directed to Chrystine Pham whose telephone number is 571-212-3702. The examiner can

normally be reached on Mon-Fri, 8:30am-5pm.

        If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q Dam can be reached on 571-272-3695. The fax phone number for the

organization where this application or proceeding is assigned is 703-872-9306.

        Information regarding the status f an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system,

see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
June 10, 2005

WEI Y. ZHEN
PRIMARY EXAMINER